ITU TECHNOLOGIES

The ITU PC Caller ID Plug

Model No. PCIDPLUG

User's Manual

Use the ITU PC Caller ID Plug to harness your computers power over your telephone!

Want an inexpensive method of capturing Caller ID on your PC? Then the ITU PC Caller ID Plug is for you!

The ITU PC Caller ID Plug is a serial-port powered device designed to deliver Caller ID data to your PC. Software for DOS and Windows with source code in 'C' and Visual Basic is included to allow your PC to act as an advanced Caller ID box. Complete information on how Caller ID works and is decoded is included to allow you to integrate Caller ID into your own applications. Your imagination is your only limit!

Features:

- Decodes Caller ID data into 1200 Baud 8-N-1serial data
- Name, Number, Date and Time is provided
- Windows 3.1, 95 and DOS software included
- Visual Basic and C source code included
- Compatible with other third-party software

Customer Information

This equipment complies with Part 68 of the FCC rules. On the top case of this equipment is a label that contains, among other information, the FCC registration number and ringer equivalence number (REN) for this equipment. If requested this information must be provided to the telephone company.
The REN is used to determine the quantity of devices which may be connected to the telephone line.

Excessive REN's on the telephone line may result in the devices not ringing in response to an incoming call. In most, but not all areas, the sum of the REN's should not exceed five (5.0). To be certain of the number of devices that may be connected to the line, as determined by the total REN's, contact the telephone company to determine the maximum REN for the calling area.

If the Serial Caller ID Module causes harm to the telephone network, the telephone company will notify you in advance that the temporary discontinuance of service may be required. But if advance notice isn't practical, the telephone company will notify the customer as soon as possible. Also, you will be advised of your right to file a complaint with the FCC if you believe it is necessary.

The telephone company may make changes in it's facilities, equipment, operations, or procedures that could affect the operation of the equipment. If this happens, the telephone company will provide advance notice in order for you to make the necessary modifications in order to maintain uninterrupted service.

If trouble is experienced with the Serial Caller ID Module, please contact ITU Technologies (513-661-7523) for repair and warranty information. If the trouble is causing harm to the telephone network, the telephone company may request you remove the equipment from the network until the problem is resolved.

No user serviceable parts or repairs may be made to this unit.

The equipment cannot be used on public coin service provided by the telephone company. Connection to Party Line Service is subject to state tariffs

INTRODUCTION

The Caller Number Identification Service, commonly known as Caller ID, has been available in most regions for several years. This module is an inexpensive solution that allows your PC, Single Board Computer or microcontroller to decode the Caller ID signals and deliver the data to the computer as 1200 Baud serial data.

MESSAGE FORMATS

Once you are a subscriber to the Caller ID service with your phone company, the Caller ID information will be sent over the telephone as a burst of data between the first and second ring. This data is sent to conform to the Bell 202 standard which calls for data to be encoded as an analog phase-coherent frequency shift keyed (FSK) serial transmission at 1200 bits per second.

Currently, two types of Caller ID message formats are used by the phone system. The single data message format transmits only the date and time of the call and the caller's number. The multiple data message format incorporates the ability for the phone company to include multiple messages in the data sent. The caller's name is usually sent using this method.

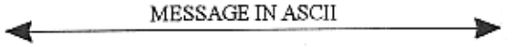
Figure 1 shows the arrangement of the message formats. The single data message format consists of a Message Type byte used to identify the message format (single data, in this case), a Message Length byte containing the number of bytes in the message following this byte and excluding the error-detection byte, eight bytes with ASCII data providing the Month, Day, Hour and Minute of the call, the ASCII Data of the Caller's Number, and finally a Checksum byte formed by taking the 2's compliment of the modulo 256 sum of the data bytes. The checksum byte can be used to detect the integrity of the data by adding the received checksum byte to the sum of the other bytes received and checking to see if the modulo 256 of this value equals 0.

The multiple message data format consists of a Message Type byte used to identify the message format, a Message Length byte containing the number of bytes in the message following this byte and excluding the error-detection byte, a series of Parameters that hold the data, and a Checksum byte as described above.

The Parameters consist of a Parameter Type, the Parameter Length, and the Parameter Data. Examples of Parameters that are used in Caller ID include Date and Time, Number, Name, Absence of Name and Absence of Number.

Caller ID Message Formats

Single Data Message Format



TYPE | LENGTH | MONTH | DAY | HOUR | MINUTE | NUMBER | CHECKSUM

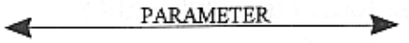
TYPE = 04h; LENGTH = 1 byte; MONTH = 2 bytes; DAY = 2 bytes; HOUR = 2 bytes; MINUTE = 2 bytes; NUMBER = data

LENGTH = Total number of bytes in message

NUMBER = ASCII bytes of number or "O" for Out of Area or "P" for Private

CHECKSUM = 2's compliment of modulo 256 sum of the message data bytes

Multiple Data Message Format



TYPE | LENGTH | PAR. TYPE | PAR. LENGTH | DATA | ...more parameters... | CHECKSUM

TYPE = 80h; LENGTH = 1 byte; PAR. TYPE = 1 byte; PAR. LENGTH = 1 byte; DATA = ASCII data

PARAMETERS:

Date and Time: PAR.TYPE | PAR. LENGTH | MONTH | DAY | HOUR | MINUTE
= 01h; 1 byte; 2 bytes; 2 bytes; 2 bytes; 2

Number: PAR.TYPE | PAR. LENGTH | Digit 1, 2, 3, ..., 10
= 02h; 1 byte; ASCII

Absence of Name: PAR. TYPE | PAR. LENGTH | "O" OR "P"
= 08h; 1 byte; "O" = out of area, "P" =

Absence of Number: PAR. TYPE | PAR. LENGTH | "O" OR "P"
= 04h; 1 byte; "O" = out of area, "P" =

Name: PAR. TYPE | PAR. LENGTH | Name Data in ASCII
=07h; 1 byte; ASCII

Figure 1. Caller ID message Formats

DESCRIPTION

Software must now be used to read the incoming data from the RS-232 serial port and decode it for proper use. The serial port must first be initialized to a 1200 baud rate, no parity, 8 bits and 1 stop bit. When this is done, the serial port can be read when data is available and the Caller ID format may be decoded as mentioned above.

The included diskette includes a program called CNID.C, a simple C program compiled using Borland C++ v3.1 for IBM PC-compatible computers, that can be used to receive and display Caller ID messages. In this program, the PC's ROM BIOS is used to handle communications with the serial port. Although this may not be the most efficient method of programming for the PC's communication ports (particularly at higher baud rates), it serves well for our example and for limited applications. The BIOS software interrupt 14h contains the basic routines needed for communications with the serial port. The functions (what is passed in the AH register to the interrupt routine) 00h, 02h, and 03h perform port initialization, read characters from the port, and return the status of the port, respectively. For more information on how these interrupt routines function, how to call and interpret the returned parameters, consult a PC reference book such as "The Programmer's Guide to the IBM PC" by Peter Norton.

The CNID.C program is straightforward and easy to follow. The init_port, read_char, and read_status functions are simply routines used to call the software interrupts discussed above. The wait_for_data functions is used to continually poll the serial port to see if new data has arrived. The functions stuff_mdmf and stuff_sdmf are used to read the Multiple Data Message Format and the Single Data Message Format, respectively, from the incoming data and fill the cnid_format data structure, which is returned by the function. The cnid_format structure consists of the members date_time, number, and name that are simply string variables corresponding to the time of the received call, the telephone number of the caller, and the name of the caller. The main function performs the task of calling the port initialization routine, detecting if data has been received and which message format it is in, calling the appropriate routine, and displaying the received data to the screen.

More advanced software, for both DOS and Windows, is included on the diskette.

INSTALLATION AND TROUBLESHOOTING

To test the Caller ID hardware, simply connect the completed circuit to the serial port of the PC, Single Board Computer or microcontroller.

Use either a terminal program set at 1200 baud 8-N-1, or the software included with the package. Please see the README.TXT file on the diskette for the latest news on the current software. We recommend the use of the Windows software for stand-alone PC use, and include the 'C' routines primarily as examples for those who wish to develop their own software or interface with other computer/microcontrollers.

Once you have the software started, arrange for another party to call your telephone number or call your number from another telephone. After the first ring, your computer should decode the caller id signal and print at least the number of the calling party and the time of day. If you are using a terminal program, you may see other characters that appear as garbage.

If the above steps do not help, or the software does recognize the hardware, you should carefully inspect your work to insure that all solder joints are good and that all components are placed properly. Use a voltmeter to check for the proper output voltages of the voltage regulators.

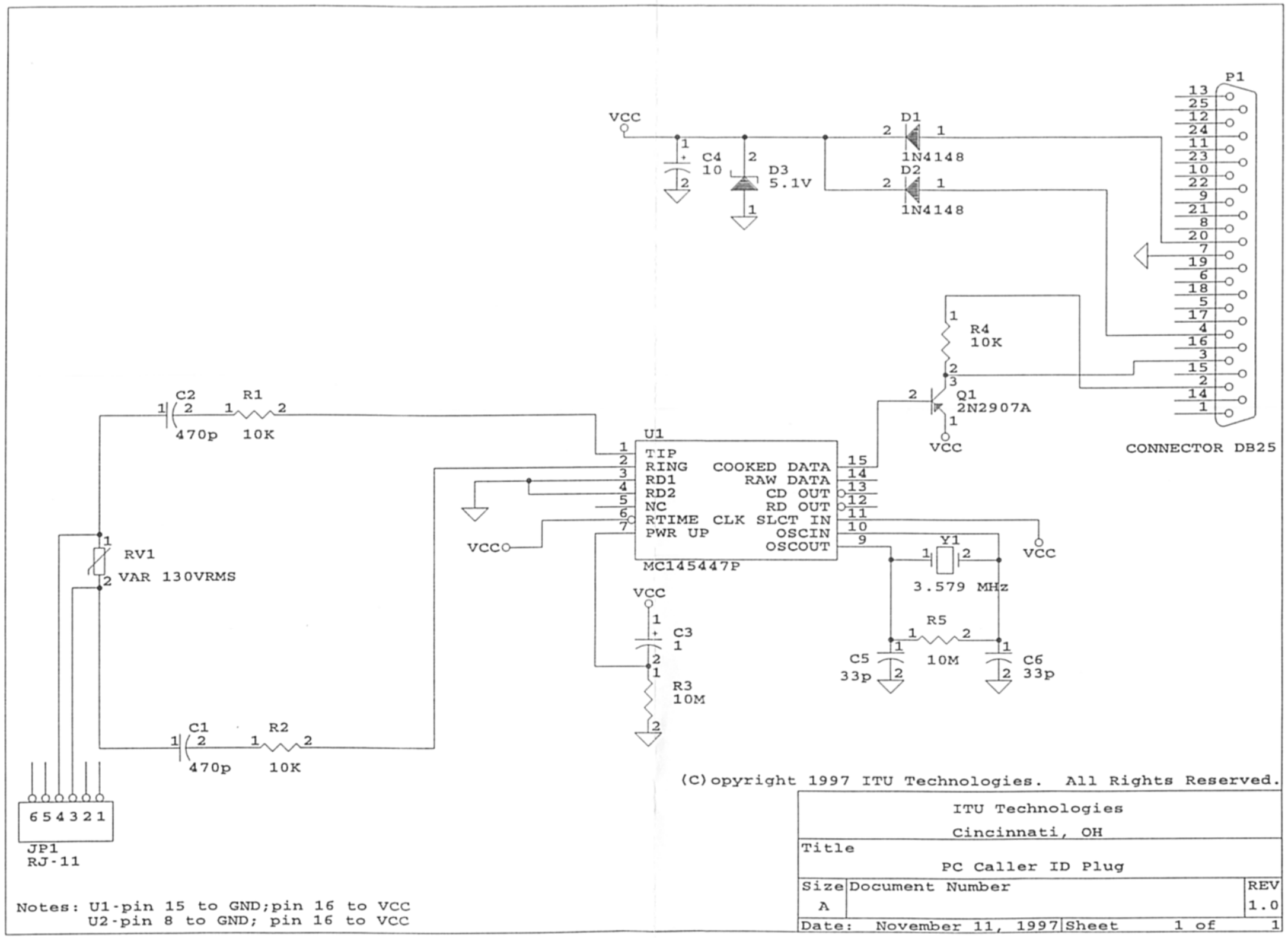
If none of these suggestions help, please phone us to see if we can readily diagnose the problem.

INCLUDED SOFTWARE

Included with this package is software to make your PC function as an advanced Caller ID Box. The README.TXT file found on the floppy includes explanations for all of the files found on the disk. Please consult this file for further information.

The programs are provided with source code to allow you to develop your own advanced Caller ID systems or to integrate Caller ID into your applications. The CNID.C program outlines the algorithm needed to decode Caller ID. This, along with basic serial port routines, can be used to write your own software in any language that you choose.

Software updates can be downloaded on our ftp site in the directory support/cnid. To log into our ftp site, simply ftp itutech.com and enter 'anonymous' at the username prompt. At the password prompt enter your e-mail address.



(C)opyright 1997 ITU Technologies. All Rights Reserved.

ITU Technologies Cincinnati, OH		
Title PC Caller ID Plug		
Size A	Document Number	REV 1.0
Date: November 11, 1997	Sheet 1 of 1	

Notes: U1-pin 15 to GND;pin 16 to VCC
U2-pin 8 to GND; pin 16 to VCC